EARLY DRAFT

Abstract Representations for Interactive Visualization of Virtual 3D City Models

Tassilo Glander[a], Jürgen Döllner[a]

[a]*Hasso-Plattner-Institute, University of Potsdam*
{tassilo.glander, doellner}@hpi.uni-potsdam.de

Abstract

Virtual 3D city models increasingly cover whole city areas; hence, the perception of complex urban structures becomes increasingly difficult. Using abstract visualization, complexity of these models can be hidden where its visibility is unnecessary, while important features are maintained and highlighted for better comprehension and communication. We present a technique to automatically generalize a given virtual 3D city model consisting of building models, an infrastructure network and optional land coverage data; this technique creates several representations of increasing levels of abstraction. Using the infrastructure network, our technique groups building models and replaces them with cell blocks, while preserving local landmarks. By computing a landmark hierarchy, we reduce the set of initial landmarks in a spatially balanced manner for use in higher levels of abstraction. In four application examples, we demonstrate smooth visualization of transitions between precomputed representations; dynamic landmark highlighting according to virtual camera distance; an implementation of a cognitively enhanced route representation, and generalization lenses to combine precomputed representations in focus+context visualization.

Key words: Generalization, 3D City Models, Landmarks, Interactive Visualization

1. Introduction

Along with an increasing demand for geographic information, virtual 3D city models – as major artifacts of 3D geoinformation – are receiving increased attention from researchers and the broader public. Driven by private companies and public municipalities, and facilitated by sophisticated technologies for the re-

mote sensing, modeling, and storing of geospatial data, virtual 3D city models (3DCMs) are evolving in both coverage and information density.

3DCMs are used in a growing number of applications that use alternative visualization strategies to communicate information connected with urban space. Non-photo-realistic rendering (NPR) depicts parts of 3DCMs in a style reminiscent of hand-drawn images in order to express planning states or uncertain details in reconstructed buildings (Maass et al., 2008; Döllner, 2007). Information visualization, such as crime mapping, relies on 3DCMs as reference frames for spatially distributed data, but can also use inherent structures as additional graphical variables. For instance, one might express the probability of a certain crime according to building or terrain height (Wolff and Asche, 2008). It seems apparent that 3DCMs are used for tasks that have traditionally been done with the help of 2D maps.

To understand how to improve visualization of 3DCMs beyond photo-realistic detail, it is instructive to look at mechanisms used in map making. As a visual medium communicating spatial information, maps have a long tradition in solving the problem of depicting large amounts of information in limited space. In cartography, the term "generalization" means the reduction of details to that necessary to achieve a given purpose by applying a set of generalization operators (McMaster and Shea, 1992; MacEachren, 1995). A related term is "schematization", which also describes simplification but aims "at cognitive adequacy [...] and intentionally distorts (aspectualises [*sic*]) a representation beyond technical constraints" (Klippel et al., 2005). Here, cognitive adequacy means that the model resembles mental knowledge representation. Both approaches require that the degree of detail being presented approximates an optimum for efficient communication.

Our article presents a method to create abstract representations of 3DCMs (Figure 1) that are focused on navigation and orientation, as needed, for example, in navigating a car. In this scenario, quick and easy perception is very important, especially if the user does not know the city. Lynch (1960) describes five major elements forming a city's mental image: paths (ways through the city, e.g., streets), edges (barriers, such as coast lines and railway lines), districts (partitions of the city), nodes (e.g., central places, path crossings), and landmarks (distinguishable objects used for orientation). Therefore, a good representation of a city has to incorporate and even emphasize these elements for easy comprehension, as well as to allow the user to connect real world structures with the displayed representations. We address this by partitioning 3DCMs into cells formed by the street network, water, railway and coast lines; this creates cell blocks, which we use as

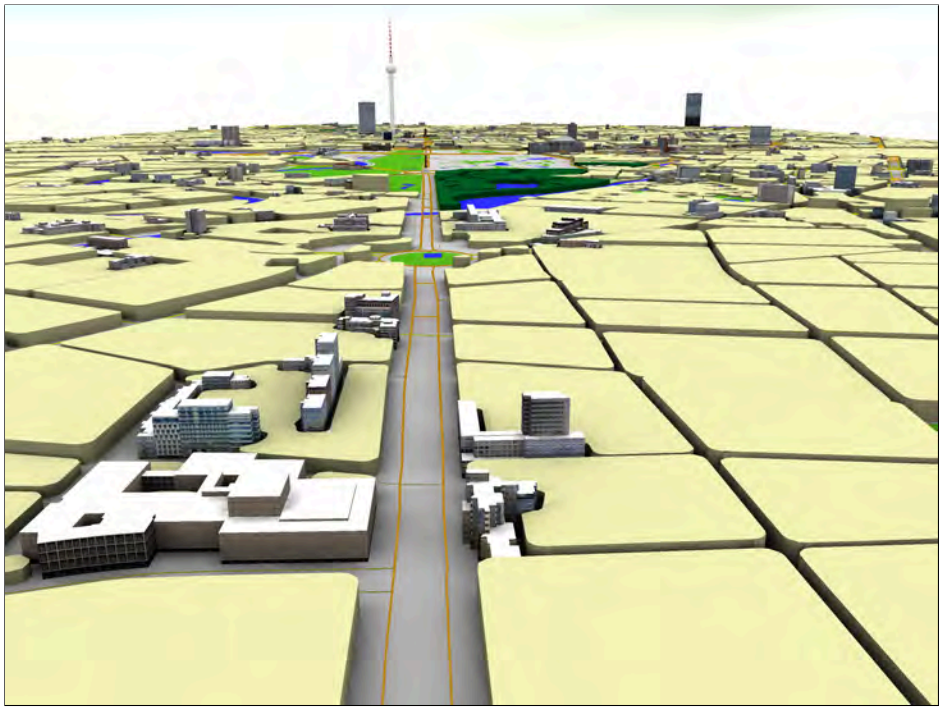


Figure 1: Abstract visualization of 3DCMs hides unimportant buildings by replacing them with simple, uniform cell blocks, while maintaining landmark buildings.

an abstract replacement for individual buildings. The road network is buffered and subtracted from the cells to emphasize paths through the city. Landmark buildings are maintained in the visualization for orientation purposes.

The remainder of this article is structured as follows. After presenting related work in Section 2, we will present the creation of abstract representations of 3DCMs in more detail in Section 3. A number of examples are sketched in Section 4 to demonstrate possible applications of our approach: we show how representations of increasing levels of abstraction can be dynamically blended in order to display an appropriate degree of detail according to distance from the viewer. Then, we present a technique for the dynamic exaggeration of landmark buildings. We also use abstracted cell blocks to implement cognitively enhanced route visualization. Finally, we present generalization lenses as a mean to integrate several abstract representations in focus+context visualization. We conclude the article with a brief discussion of our technique (Section 5), before pointing to further improvements and open questions in Section 6.

2. Related Work

2.1. Abstract Visualization of 3D City Models

Research from many fields describe 3DCMs in the context of information visualization, where the focus is on communication rather than photo-realistic rendering.

In the work of Grabler et al. (2008), a system for the creation of static tourist maps is presented. The authors introduce and apply to the model a complex metric to measure the semantic, visual, and structural importance of elements of the city in question. For the creation of city maps, they integrate a number of techniques for building simplification, optimization, displacement, and labeling. This approach creates pleasing but static oblique images of a city, while we focus on interactive 3D visualization.

Döllner (2007) discusses 3D geovisualization and 3DCMs in the context of cartographic information display. The author emphasizes the use of non-photo-realistic (NPR) styles to create expressive, illustrative visualization of 3D virtual environments.

Chang et al. (2007) present a focus-dependent visualization of 3DCMs that maintains the legibility of the city. The authors cluster buildings using a distance metric and simplify the 2D hull that wraps building footprints within each cluster. The simplified hull is then extruded to the weighted average height of the buildings it contains. The paper concentrates on visualizing urban statistical data

upon an abstracted 3D representation of the city together with 2D information visualization tools, e.g., parallel coordinates plot.

Generalization strategies for building models based on road cell aggregation, volumetric simplification, and Delaunay triangulation are presented by Glander and Döllner (2007); their study aims to reduce the complexity created by a large number of single buildings. Since only models of buildings are incorporated in the simplification process, employing the resulting visualization in applications such as tourist maps is of limited use.

CityGML, an exchange standard for 3DCMs by the *Open Geospatial Consortium* (OGC, 2006a), allows storing building geometry in four levels of detail (LOD). This is limited to generalization regarding single buildings and is therefore not covering large changes of scale. In addition, the standard defines a very flexible generalizes-to relation to link arbitrary members of a 3DCM. However, it does not specify a procedure of how to create these generalized representations.

2.2. 3D Generalization Techniques

In the context of 3DCMs, 3D generalization has been used mainly for rendering terrain and simplifying single buildings. In both instances, level-of-detail (LOD) representations are created to reduce computational complexity and improve rendering speed.

Terrain geometry can be easily generalized for smooth, view-dependent rendering, as connected triangular irregular networks (TIN) and grid structures can be simplified using many techniques (Pajarola and Gobbetti, 2007). However, generic simplification techniques, such as those surveyed by Cohen and Manocha (2005), typically perform badly on building geometry, since the majority of buildings are already low-polygon objects. Additionally, specific properties such as parallelism and orthogonality need to be respected by simplification techniques. Therefore, specific 3D building generalization techniques take into account such properties during simplification (Kada, 2007; Forberg, 2007; Rau et al., 2006). Generalization of groups of buildings, such as that shown for linear building groups, is seldom addressed (Anders, 2005). However, for abstract visualization of 3DCMs containing hundreds of thousands of (individually simple) buildings, aggregation is essential to hide unnecessary complexity.

Compared with map generalization techniques in 2D, generalization in 3D is still in its infancy. 2D map making has been formalized as a process in which one must apply a number of generalization operators that are potentially conflicting with each other (McMaster and Shea, 1992). After the development of single 2D operators, recent approaches integrate several operations in frameworks that try to

resolve occurring conflicts and enforce defined constraints (Barrault et al., 2001; Lamy et al., 1999; Sester, 2000).

3. Cell-Based Generalization

In this section, we describe our technique, which creates abstract representations of 3DCMs. This is done once, prior to those applications presented in Section 4.

In the following, we refer to a 3DCM that is described by a 3-tuple $CM = (B, I, A)$ with a set of distinguishable building objects B, having geometry plus optional façade textures, a set of infrastructure elements I, given as line strings, and a set of optional non-building areas A, given as polygons. We need two weighting functions to be defined for the tuple: $line\_weight : I \rightarrow \mathbb{R}^+$, weighting each infrastructure element, and $building\_weight : B \rightarrow \mathbb{R}^+$, weighting each building according to its qualification as a landmark.

This technique can be described as mapping the given CM to subsequent representations of increasing levels of abstraction LOA. We identify the input model with the lowest *level of abstraction* $LOA_0 := CM$ and define the mapping function $generalize(LOA_{i-1}, i) = LOA_i$. We use the term "level of abstraction" instead of "level of detail" (LOD), since LOD is typically connected with simplification motivated by computational requirements, as opposed to that required to reduce cognitive effort.

3.1. Input Data

Buildings B for 3DCMs can be divided into two classes. Today, the majority are still prismatic 2.5D buildings derived from cadastral databases, where building footprints and heights are maintained together with additional attributes, such as construction date or ownership. While they are geometrically simple, these buildings typically cover the whole city area and sometimes also have façade data attached. This class of buildings is read by our technique as shape files, encoding polygons and related attributes[1].

In addition, a smaller number of buildings are modeled in higher detail. While sometimes these models already have been created for newly constructed buildings, existing buildings still have to be modeled manually or derived from high resolution scanning, which restricts this method to a few, typically important

[1] http://www.esri.com/library/whitepapers/pdfs/shapefile.pdf

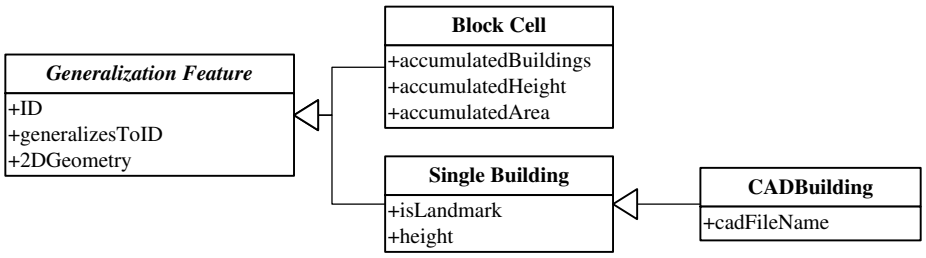


Figure 2: Generalization features are the central unit during the generalization. They can be aggregated block cells or single buildings.

buildings. For automatic processing, object identity information needs to be available to distinguish single buildings within the models. Standard formats, such as 3DS, COLLADA or KML, as exported by CAD tools, or CityGML are read by our software.

To be able to process the two different types of buildings similarly, we convert them to a simple internal data model (Figure 2): one *Generalization Feature* is the smallest unit that our technique processes. This unit can be either a building (CAD or 2.5D), or a generalized block cell. It holds a single polygon, an *ID* value and a *GeneralizesToId* value to enable us to track the generalization relation between features. For block cells, statistics such as the number of grouped buildings are also specified. For CAD models, the projected footprint is stored together with the individual height of their bounding box and a reference to the original geometric model.

Infrastructure data I consist of line features that are weighted. These can be provided by commercial data providers, such as Teleatlas or Navteq, or by the community — through OpenStreetMap, for example. While TeleAtlas data differentiate nine levels of importance for roads, OpenStreetMap data provide about five main levels, but additionally include small bicycle and pedestrian routes.

In addition, non-building areas A, such as land use and land coverage data, can be added to integrate green spaces, lakes or wooded areas. For these natural features, OpenStreetMap data have a greater richness, at least in big cities with active, contributing communities.

3.2. Cell Processing

The basic idea is to replace multiple single buildings by a single block cell enclosed by infrastructure elements. To do this, the input infrastructure network together is intersected with the contours of non-building polygons in order to create an arrangement containing polygons (Berg et al., 2000). These polygons are further processed by cutting out buffered streets and non-building polygons. To remove small sliced polygons introduced by Boolean operations, morphological closing is performed by consecutively shrinking and growing the polygons. So far, the results are cell polygons similar to 2D settlement areas.

In the next step, buildings are mapped to those cell polygons they overlap most; then the mean height and variance for buildings of a single cell are computed and recorded. Mean height will be used later to extrude polygons to 3D cell blocks, to communicate the average building height of the abstracted buildings. This mapping is stored by setting a unique *ID* value and a *GeneralizesToID* value for generalized items.

Additionally, other attributes of buildings within each cell can be evaluated and aggregated. For example, semantic attributes such as statistical data about rental fees or building usage can be averaged to yield an abstracted value for the entire cell block, substituting for values for individual buildings contained therein.

This process is repeated, removing the least important streets, as defined by *line_weight*, in each iteration. Hence, *LOA* geometry is created from scratch in each iteration, as cells are created by computing a new arrangement. Resulting generalization features from the previous iteration are fed into the current one to map them to the newly created cell blocks. Thus, cells grow fewer and larger while a generalization hierarchy is created, traceable by the *generalizes-to* relation (Figure 2).

3.3. Handling Landmark Buildings

In a spatial environment, landmarks are essential elements for navigation and orientation. They represent distinctive objects having key characteristics that cause them to stand out from their surroundings (Sorrows and Hirtle, 1999). In virtual environments, landmarks are especially important, since – compared with real environments – fewer spatial and locomotive cues are presented to users as they move through the environment. Explicitly designing landmarks in virtual environments such as 3DCMs therefore can facilitate navigation and the acquisition of spatial knowledge (Vinson, 1999).

To identify landmark buildings, our technique relies on an externally defined weighting function *building_weight*, which represents an individual building's im-

portance as a positive real number. A number of approaches discuss the detection of landmark buildings by assessing their saliency: integrating visual, semantic, and structural properties in relation to neighborhood. See, for example, (Grabler et al., 2008; Raubal and Winter, 2002; Winter et al., 2008).

If a weighting function cannot be provided as input data, we apply a simple mechanism using relative height as an important aspect of visual legibility to detect local landmarks (Sorrows and Hirtle, 1999): once the mean height $\overline{h}$ is computed in the previous step, buildings that deviate significantly from their cell neighbors' height are identified as local landmarks and put into the set of initial landmarks. We describe "significantly" in k units of the standard deviation σ, typically with $k = 1.5$.

$$isLandmark(b_i) = height(b_i) > k \cdot \sigma + \overline{h} \tag{1}$$

Additional buildings may be selected manually for the initial set of landmarks, for example, points of interest such as churches.

3.4. Creating a Landmark Hierarchy

For higher *LOA* representations, we use a different technique to identify landmarks. The goal is to reduce the number of landmarks, while keeping important ones and maintaining an even distribution. For example, a newspaper shop may be used as a landmark in the surrounding neighborhood, but is rather unimportant compared to the whole city. Visualizing it would be essential in a large scale context, but regarded as noise in small scale context.

Simply choosing a higher deviation factor k in Equation 1 does not lead to a desirable landmark selection, as k cannot be determined automatically for different data sets; nor does it yield a balanced distribution. Instead, we found that the algorithm adapted from Winter et al. (2008) worked nicely as follows:

- Triangulate landmarks using their centroids for a Delaunay Triangulation (DT) (Berg et al., 2000).
- Compare each vertex' saliency with direct neighbors and vote for the highest saliency vertex among them.
- Promote vertices that have at least one vote to the next level.

As stated before, the concept of saliency includes visual, semantic and structural properties (Sorrows and Hirtle, 1999). However, we refer the problem of

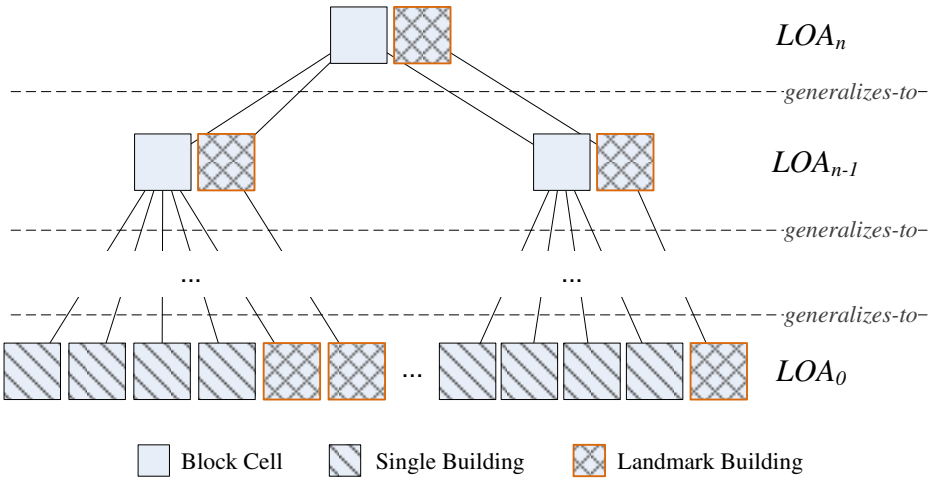


Figure 3: Cell hierarchy defined by the *generalizes-to* relation is aligned with the landmark hierarchy.

computing saliency to the *building_weight* function, and simply use height if a more sophisticated measure cannot be provided.

In the resulting landmark hierarchy, the number of vertices and the corresponding landmark buildings is steadily reduced in subsequent layers of the hierarchy. On average, we found a reduction factor of about three. The algorithm stops when just one landmark is left. A nice property of this method is that the iterative elimination of landmarks in higher hierarchy levels maintains an even spatial distribution. In addition, it requires no manual interaction.

The landmark hierarchy is then aligned with the previously created cell polygons (Figure 3): the most abstract computed representative LOA_n is correlated to landmark buildings of the highest level of the landmark hierarchy, the next LOA_{n-1} uses landmarks from the second level of the hierarchy, and so on.

For each representative *LOA*, the landmarks have to be removed geometrically from the containing cells by cutting them out. Similarly, mean height calculation is recalculated, as landmark buildings no longer contribute to the block cell. Instead, they are included unchanged in the *LOA* representation – that is, no simplification is applied to them.

3.5. Handling Non-Building Elements

Non-building elements include green spaces, wooded and water areas. As they do not contribute to built-up areas, they are cut out of block cells, using Boolean

(a) Polygon manipulation (b) Wooded area close-up (c) Wooded area total

Figure 4: Abstract representations of wooded areas are created by adding and moving additional points on polygon (a) and extruded surface (b). Contained water areas are cut out of the polygon (c).

operations. Green spaces, park areas, and water areas are colored accordingly and added to the scene.

For wooded areas, we create three dimensional representations to communicate the fact that they effectively occlude objects behind them. In contrast to photo-realistic visualization, which require detailed vegetation models, we need an abstract representation. In 2D maps, signatures are used to abstract from specific structures and, for example, distinguish built-up areas and vegetation areas (MacEachren, 1995). We construct an abstract 3D representation by inserting additional points along polygon edges, which are then moved randomly within a radius to create a fuzzy look (Figure 4). After extruding the polygon to typical tree heights, the top surface is also "frazzled" by randomly adding points according to a Poisson distribution.

The road network, as a major non-building element, can also be added to the scene. While roads are already visible implicitly through space between block cells, adding them with color and width according to individual road weight further enriches visualization.

3.6. Results

For rendering the resultant geometry, we use the OpenGL-based open source scene graph library VRS[2]. To enhance depth cues, shadow textures are precomputed using an ambient occlusion technique (Döllner et al., 2006; Landis, 2002). Our viewer application allows switching between different static *LOA* representations.

[2]The Virtual Rendering System, www.vrs3d.org

	# buildings (app.)	# streets (app.)	# road weight classes
Berlin	65.000	13.000	9
Berlin (small)	14.000	1.700	8
Boston	150.000	17.000	5
Cologne	150.000	6.000	5

Table 1: Comparison of the four datasets used.

LOA	0	1	2	3	4	5	6	7
# generalization items	14.000	585	583	206	95	43	23	13
# landmark buildings	-	295	295	105	39	9	4	1

Table 2: Number of items for all *LOA* representations for the small Berlin data set. Landmark buildings are a subset of generalized items. As the landmark hierarchy has just six levels, LOA_1 and LOA_2 have the same set of landmark buildings.

We applied the cell-based generalization technique on the 3DCMs of Berlin, Boston and Cologne for evaluation. While we used only simple buildings and OpenStreetMap data for Boston and Cologne (Figure 5), we added façade textures, additional CAD building models, and TeleAtlas road data to the 3DCM of Berlin (Table 1).

Evaluating the results for these datasets shows that cell-based generalization yields representations appropriate for an abstract visualization. However, the assumption that individual buildings can be substituted by a cell block built by surrounding streets fails for sparsely built areas, such as the peripheral areas of Cologne and Boston. If these spaces have not been specified as non-building areas, the resulting representation is misleading. Still, using average height to determine the extent of extrusion for abstracted blocks reflects the general appearance of a 3DCM and allows, for example, to distinguish downtown areas from suburban areas.

Since the generalization process is based on the road network, a natural hierarchy of abstract blocks can be created using road weights (Figure 6). Table 2 shows that for the small Berlin dataset, the number of single items is reduced at each level, compared to the previous one. As the computed landmark hierarchy has levels fewer than the total *LOA*s, the number of landmarks does not change from LOA_1 to LOA_2.

The presented results are static, abstract representations of 3DCMs that can be explored interactively. They fulfill the goal to hide complexity while maintaining landmark buildings and important structural relations necessary for orien-

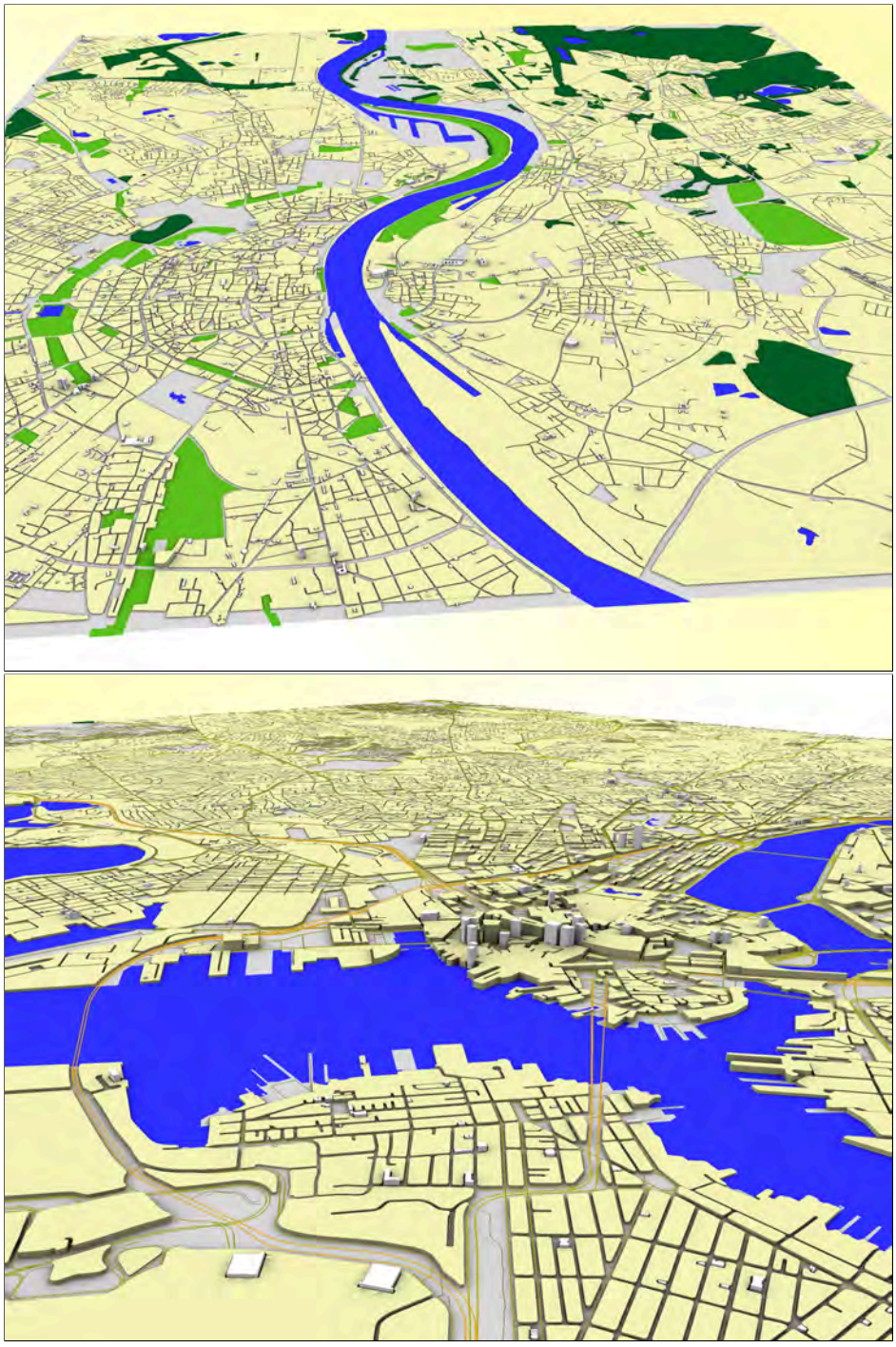
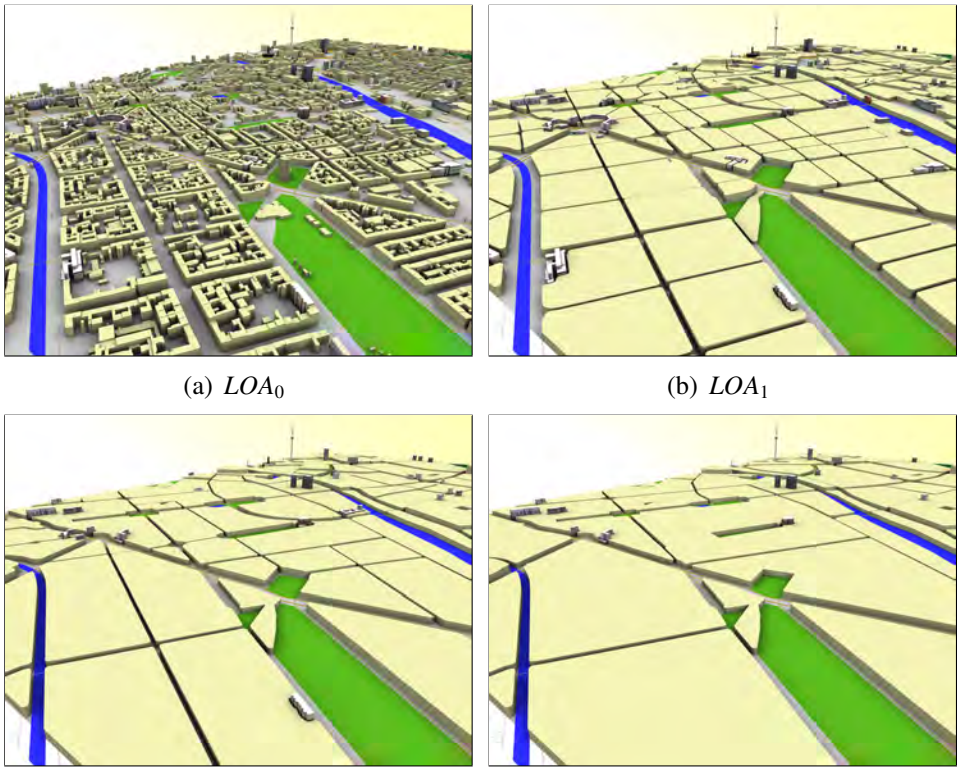


Figure 5: The technique was tested on city models of Cologne (top) and Boston (bottom).



(a) LOA_0 (b) LOA_1

(c) LOA_3 (d) LOA_4

Figure 6: For increasingly abstract representations, the algorithm creates increasingly large block cells with decreasing numbers of landmark buildings (small Berlin dataset).

tation and navigation. In addition, once these discrete representations have been precomputed, a number of applications can be derived that further exploit interactivity, regard cognitive principles, or create focus+context visualization. They are sketched in the following.

4. Applications

4.1. Multi-Scale Rendering with Smooth Transitions

When 3DCMs are visualized, a perspective projection is typically applied during rendering (Akenine-Möller and Haines, 2002), creating a view matching real world experience: objects far away are perceived as small, and grow larger as the viewer approaches them. In other words, rendered 3D views are perceived in one continuous scale, as is the real world. Classical maps, by contrast, typically show a top down view on single scale.

When presenting multiple scales, one approach would be on-the-fly generalization to obtain the desired level of abstraction. However, due to the high computational complexity of generalization techniques, typically a number of generalized representations will be precomputed at different scales (Cecconi and Galanda, 2002). Interactive 2D maps provide the ability to change scale by switching between these precomputed representations. More sophisticated approaches render *transitional* maps when changing scale between precomputed *stationary* maps to give the impression of smooth zooming (van Kreveld, 2001), as shown for buildings (Sester and Brenner, 2004), polylines (Nöllenburg et al., 2008), and between arbitrary shapes (Danciger et al., 2009).

In 3D, progressive meshes simplify geometry and provide smooth interpolation for terrain models (Hoppe, 1998). Unfortunately, typical simplification methods do not work well for large numbers of single models with individually low polygon counts, such as buildings.

Therefore, we present a strategy to render 3DCMs in multiple *LOA* representations as computed before (as described in Section 3) and display them based on distance from the virtual camera. We propose transition mechanisms based on vertical motion and transparency.

4.1.1. Validity Ranges

With multiple representations available, one needs to decide which representation should be shown for a certain scale.

Validity Ranges describe the validity of objects of a certain representatative LOA_i, i.e., generalization items, as a tuple of distances to the virtual camera

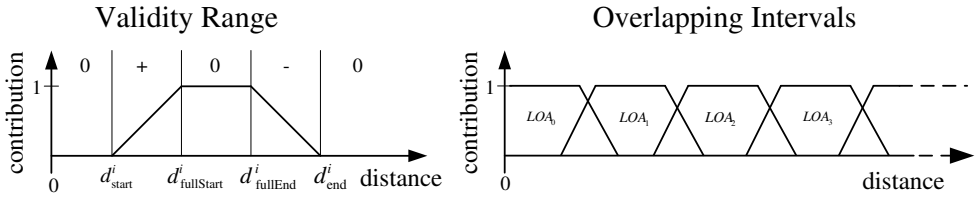


Figure 7: For each *LOA* representation, a validity range is set up by four distances (left). Overlapping intervals enable smooth transitions (right).

$(d^i_{\text{start}}, d^i_{\text{fullStart}}, d^i_{\text{fullEnd}}, d^i_{\text{end}})$. While the interval $[d^i_{\text{start}} : d^i_{\text{end}}]$ describes, when a *LOA* representation becomes visible and invisible, respectively, the interval $[d^i_{\text{fullStart}} : d^i_{\text{fullEnd}}]$ specifies the range, within which it is completely visible (Figure 7). A non-overlapping partition of the view axis into distance intervals $[d^i_{\text{fullStart}}, d_{\text{fullEnd}}]$ would imply hard switches of *LOA* representations. Therefore, some overlapping of intervals is needed to create a visual transition effect. Interval boundaries can be specified individually for each *LOA* representation or can be set automatically according to a linear or logarithmic mapping for a given maximal distance.

Given an interval setup, a $contribution_i$ function defines, for each generalization item of LOA_i and a certain camera distance, a normalized contribution factor and a change direction $contribution_i(d) : \mathbb{R}^+ \mapsto (0 \leq x \leq 1, \{+, -, 0\})$. The contribution factor is computed as follows:

$$contribution_i(d) = \begin{cases} (0,0) & d \leq d^i_{\text{start}} \vee d > d^i_{\text{end}} \\ (\frac{d - d^i_{\text{start}}}{d^i_{\text{fullStart}} - d^i_{\text{start}}}, +) & d^i_{\text{start}} < d \leq d^i_{\text{fullStart}} \\ (1,0) & d^i_{\text{fullStart}} < d \leq d^i_{\text{fullEnd}} \\ (\frac{d - d^i_{\text{fullEnd}}}{d^i_{\text{end}} - d^i_{\text{fullEnd}}}, -) & d^i_{\text{fullEnd}} < d \leq d^i_{\text{end}} \end{cases}$$

Because of overlapping intervals, it is possible that certain places in a 3DCM be covered by multiple geometries at the same time.

4.1.2. Transparency Blending

Having computed contribution factors, a straightforward transition mechanism would be to blend different *LOA* geometries using transparency. Transparency blending mixes a source color with a new color based on a blending equation, typically controlled by an alpha value that specifies the opacity for each color value. For visualizing transitions between *LOA* representations, we use the calculated

Figure 8: 3DCM representations are blended using transparency.

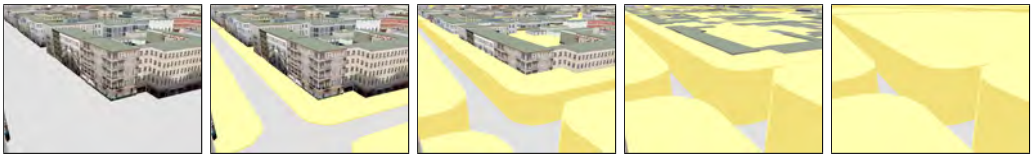


Figure 9: By vertically scaling *LOA* representations, a smooth transition is obtained.

contribution factor as alpha and let OpenGL do the blending. As the contribution factor is determined for every rendered image according to virtual camera distance, a dynamic blending effect is achieved when moving through the 3DCM (Figure 8).

4.1.3. Vertical Moving

An alternative transition effect can be obtained using vertical movement of generalization items. When items become visible, that is, $contribution_i = (x, +), 0 < x < 1$, they will move into the scene by applying vertically scaling, using the contribution value x directly as a scaling factor. When becoming invisible, that is, $contribution_i = (x, -), 0 < x < 1$, items are scaled to the height of the subsequent generalization item, using the *generalizes-to* relation to achieve smoothness (Figure 9).

4.1.4. Results

Multi-scale rendering of 3DCMs yields a visualization that handles varying information density throughout the depicted scene. Dynamic transitions between *LOA* representations make navigation within the model smooth and help to maintain coherence while the visual geometry changes (Figure 10). Therefore, interactive visualization of a 3DCM is enhanced by dynamically adapting information density while exploring it.

As an extension of this technique, it is possible to select a building of interest and use its position for reference. Starting from the highest *LOA*, cells are subdivided, or replaced by the objects they generalize, if they contain the given

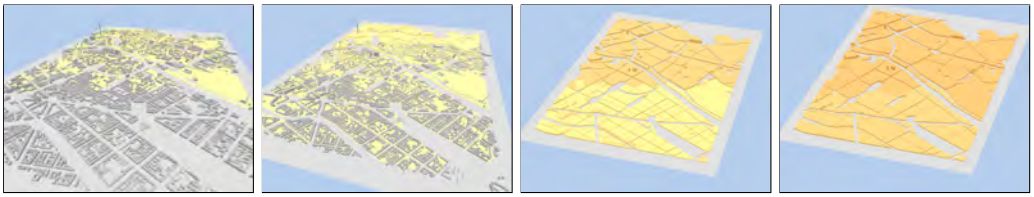


Figure 10: Increasing camera distances are simulated (left to right) to show the effect of transitions between several *LOA* representations. Landmark buildings and non-building areas are not regarded by this technique.

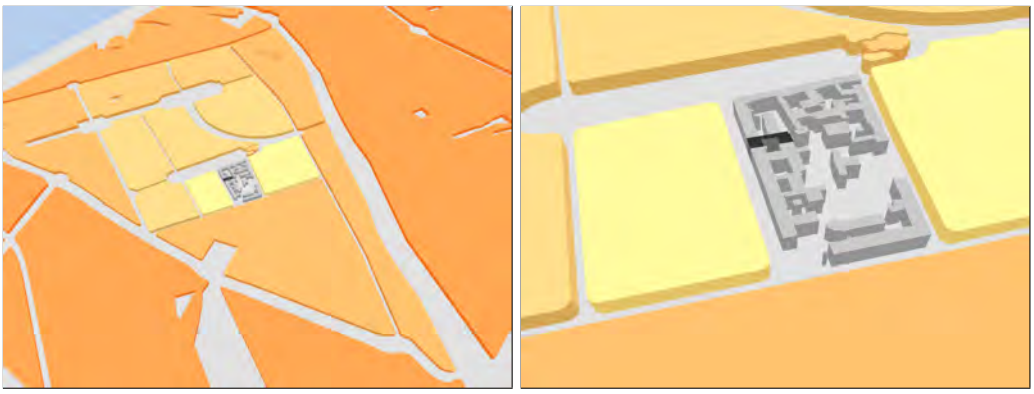


Figure 11: A selected building (black) is presented in its surrounding blocks by gradually decreasing details towards the context region.

building. With growing distance from the reference position, the level of abstraction increases (Figure 11). This way, contextual information is still given, but it does not distract from the building in question.

Another extension handles transition maps after camera movement has stopped. It can be argued that transition maps should only be shown during movement, as they merely represent intermediate states (van Kreveld, 2001). Therefore, our technique can be configured to perform a "snap back" of generalization items to their nearest stationary form once camera movement has stopped.

4.2. Dynamic Landmark Highlighting

2D city maps for tourists typically show single residential buildings as simple blocks while emphasizing interesting buildings, such as landmarks, by highlighting them for easy apprehension. While maps have different visualization styles for landmark buildings, ranging from textual to iconic to realistic (Elias and Paelke, 2008), for virtual environments realistic representations are preferred (Vinson,

1999). Therefore in our second application example, we present a technique that dynamically enlarges landmark buildings within 3DCMs depending on the virtual camera distance when they are highlighted. To reflect the fact that, in reality, an individual landmark is referred to only in a certain region (Winter et al., 2008), we limit highlighting to a distance interval that depends on each landmark's approximated relevance.

4.2.1. Scale Factor

The goal of highlighting landmark buildings is to make them more visible. Usually, perspective projection scales objects in 3D environments, as objects are projected on the 2D image plane, which leads to an inversely proportional relation between camera distance and projected size (Akenine-Möller and Haines, 2002). Therefore, our highlighting requires for each landmark an individual scaling function $scale_i(d)$ that compensates for the downscaling of distant objects caused by perspective projection. To conform to the idea of the spatially limited relevance of a landmark, we set for each landmark building a distance interval within which it is dynamically rescaled.

The interval $I = [d_{\text{start}} : d_{\text{end}}]$ describes a range where the projected size should be kept constant (Figure 12). This is done by using camera distance as a scaling factor. When the landmark leaves the interval, exaggeration is stopped and the building reverts to its original size. For a smooth transition, we use a quadratic function and calculate its coefficients according to the interval I.

Starting distance describes the distance from which the projected size of the landmark should remain constant while zooming out. We use $d_{\text{start}} = 3000$ m for all landmarks to yield a projected height of 20 to 50 pixels, depending on the buildings' size and the display window's size; this value leaves objects recognizable. It is also possible to directly specify the desired pixel height.

To create ending distance parameters for exaggeration, distance values can be specified per landmark class – that is, depending on their highest level in the landmark hierarchy (Section 3.4).

4.2.2. Displacement Handling

To account for enlarged landmark buildings, in general a displacement strategy has to be applied to avoid intersecting geometries. In our application, we distinguish surrounding highlighted landmarks and other geometries from cell blocks, land use polygons and infrastructure elements.

Other geometry such as abstracted cell blocks in which highlighted landmarks are situated are less important and, therefore, can be sacrificed in favor of more

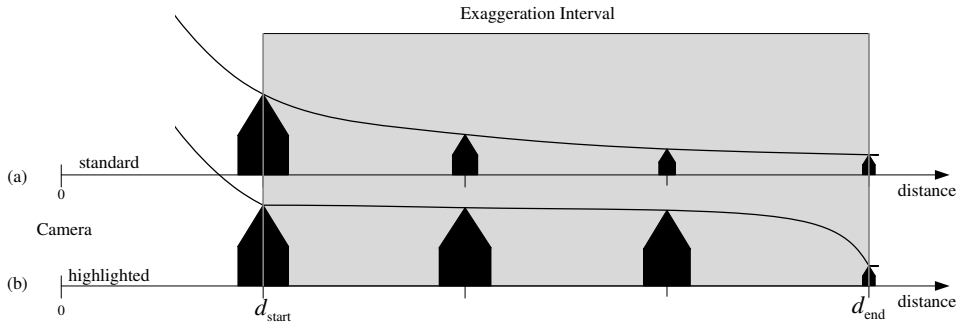


Figure 12: Traditional perspective projection (a) scales objects inversely proportional to distance. Our scaling (b) keeps the size of the landmark object nearly constant within a distance interval and returns it to original size at the interval limits.

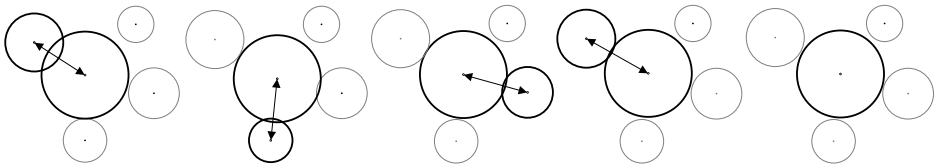


Figure 13: The example shows how the displacement works: overlapping landmarks are shifted in regard to each other iteratively. This is repeated until no more overlapping occurs or a maximum of iterations is reached.

important highlighted landmarks. Hence, we simply clip all other features of the scene against the growing radial distortion zone of highlighted landmarks. This simplification, however, works only for a limited scale range, as it may happen that exaggerated landmarks cross surrounding roads or coast lines, which should be avoided.

Other highlighted landmarks need to be displaced by performing collision handling. Given the small number of highlighted landmarks, a naive displacement works: each landmark is approximated by its 2D bounding circle and tested against all others. If overlapping occurs, both landmarks are slightly moved away from each other, which resolves the single collision, potentially causing a new one. This is iterated until no more collisions occur or a maximum number of iterations is reached (Figure 13). This is done for each rendered image.

4.2.3. Results

Dynamic highlighting of landmarks provides a visual model that reveals main landmark buildings by scaling them to a visible size (Figure 14). As their visibility is enhanced, landmark objects can be used for orientation, as in reality, but from a greater distance, amplifying their impact.

Dynamic highlighting should still be limited to a selected number of the most global landmarks for several reasons. If too many objects were emphasized this way, highlighting would lose its value, as the display would be cluttered with too many items. In addition, scaling landmarks are perceived as movement, which captures the user's attention. Therefore, it should be used sparsely. Finally, due to technical limitations, scaling many objects would degrade real-time performance because of the needed displacement operations.

The visual model could be enhanced considerably by better displacement handling where impenetrable surrounding geometry, such as streets and water areas, would be respected. For real-time rendering, however, approaches used in collision handling (Möller and Haines, 1999) may be preferable when compared to classical map displacement techniques, as discussed in (Lonergan and Jones, 2001), as computation needs to be efficient enough to maintain interactive frame rates.

4.3. Cognitive Studies

Orientation and navigation in 3D virtual environments has been an ongoing research question for many years. Research from cognitive science shows that route perception and processing can be facilitated if its visual model is adapted to correspond with mental concepts of a route. Cognitively, a route is a sequence of decision points – such as road junctions – which are usually remembered in their prototypical configurations (Klippel et al., 2005). For example, asked for a route description to find the way to some place in a city, most people will give a list of turning actions: "Go straight ahead, then turn sharp left, then right,..." Especially, they will probably not describe the exact turning angles at decision points, but will resort to a few prototypical directions.

An exemplary application of these so called wayfinding choremes[3] has been proposed for 2D maps by altering route junctions (Klippel et al., 2006). Therefore, we implemented concepts for 3D wayfinding choremes in a 3D virtual environment to be able to evaluate them in user studies (Glander et al., 2009). Our

[3] *chor* - greek word for space, elementary primitives of space

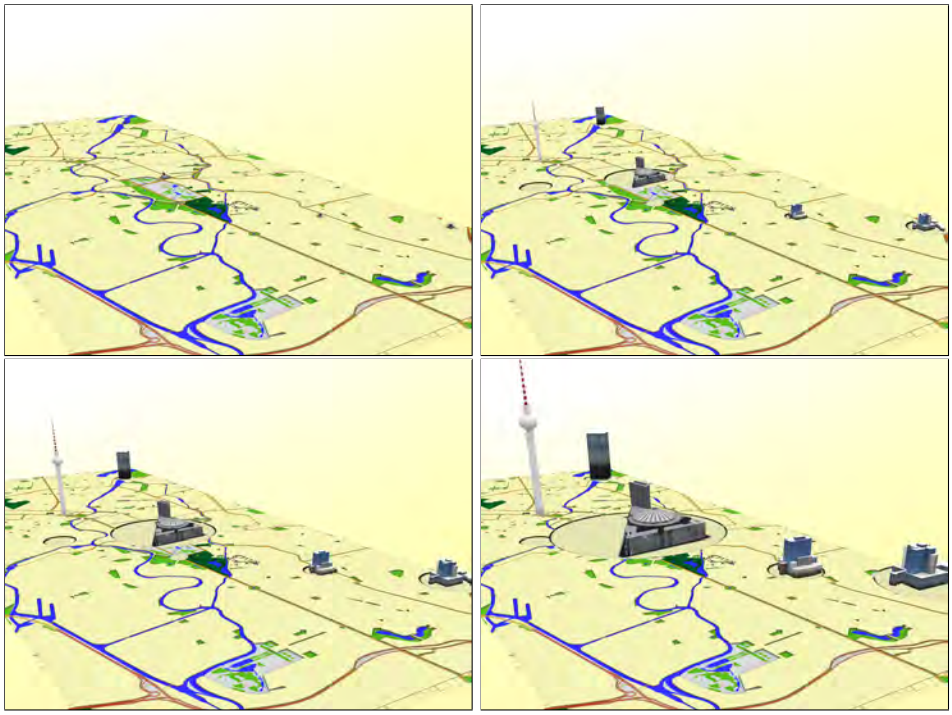


Figure 14: Traditional perspective projection (upper left) scales objects inversely proportional to distance. Our scaling (upper right) keeps the size of the landmark object nearly constant within a distance interval and returns it to original size at the interval limits. Starting distance can be set for bigger exaggeration (lower left), up to pictorial visualization (lower right).

abstracted block cells are good candidates, as they are derived from roads and directly reflect changes in a road's course. In addition, they are sufficient to run an experiment without needing to displace or deform individual buildings along roads.

4.3.1. Choreme Processing

For this application, only a set of roads and a route along them are needed. Starting with the first route node, our technique iterates through all junctions on the route, transforming them: for each junction, the incoming road is set as the reference direction, and all outgoing roads are sorted into eight bins reflecting the prototypical directions in 45° steps (eight-sector model). If no unique mapping can be done – that is, more than one road is put into one bin – the junction is left unchanged.

Each outgoing road is then transformed by rotating it to the prototypical direction. New points are inserted within a given radius, and the old connection points are connected using Bezier curves for a smooth connection (Figure 15).

The transformed road network is then polygonized and buffered roads are cut out, as described in Section 3.2. However, to preserve sharp features, which are needed to communicate configurations of junctions, no morphological closing is applied. For a 3D virtual environment, the resulting polygons are extruded to cell blocks, and façade images are applied randomly to create the look of a city.

4.3.2. Results

We developed an application suitable for conducting experiments including joystick interaction, collision handling, position tracking, and monitoring. Experiments are currently being done to evaluate 3D wayfinding choremes, comparing subjects' navigation performance in an unchanged environment with a cognitively enhanced environment.

Abstracted block cells have proven suitable especially for this user study, as they allow the creation of a defined environment where unwanted influences, which might interfere with the research question, can be excluded. For example, to evaluate wayfinding choremes, other navigation aids, such as uniquely shaped buildings or significant façades that may serve as landmarks, can be excluded from the visual model. The abstraction in this application example is extended to route decision points – road junctions – which are adapted to their prototypical configuration.

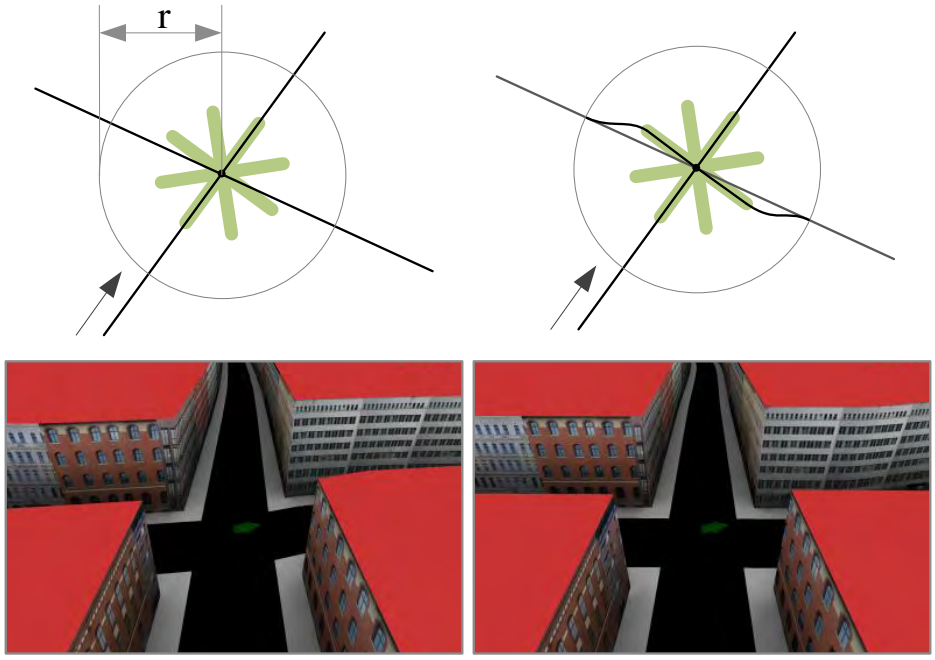


Figure 15: Schematic and 3D example of a junction before (left) and after (right) processing of road segments. The radius *r* can be specified to limit the adaptation effect.

4.4. Generalization Lenses

Once distinct geometric representations of 3DCMs are computed, focus+context visualization can be obtained using generalization lenses. Magic lenses (Bier et al., 1994) reveal hidden structures without necessarily magnifying content. We implemented a visualization technique based on volumetric lenses to integrate several *LOA* representations in one image (Trapp et al., 2008).

4.4.1. Lens Creation

Arbitrary 3D lens shapes are first converted into a voxel representation, and stored in layered depth images called *volumetric depth sprites* (VDS) for efficient rendering (Trapp and Döllner, 2008). Lens shapes can either be modeled explicitly using 3D modeling software, or derived from buffered 2D polygons and polylines. By applying a transformation to VDS, interactive rotation, scaling, and translation of lenses is possible. To support multiple, possibly overlapping lenses, each lens is mapped to a *LOA* representation and assigned a priority.

4.4.2. Lens Rendering

During rendering, pixel-precise clipping is done within a single pass to determine to which *LOA* representation a fragment belongs, in order to draw it accordingly.

The algorithm starts by rendering the context geometry – the geometry associated to the space surrounding all lenses – clipped against all other lens volumes. Then, starting with the lowest priority, *LOA* geometry is rendered successively within its respective lens volume, thereby clipping it against all lens volumes with a higher priority. This is done until all lenses are rendered.

4.4.3. Results

The lens technique allows for the flexible combination of different *LOA* representations within an efficient rendering framework. Use cases include the highlighting of focus regions and routes within 3DCMs (Figure 16). Since lens volumes can be moved in real-time, also dynamic regions of interests – such as the area around the current position of a car driver – can be highlighted. A limitation of the approach occurs in the resolution of the depth images used to represent lens volumes. Depending on the chosen resolution, staircase artifacts can occur. This application example demonstrates interactive visualization of abstract representations in a complementary way to the multi-scale rendering technique presented before.

Figure 16: Generalization lenses allow the combination of different *LOA* representations in one image. Therefore, for example, focus regions of arbitrary shape (left) or the course of a buffered route, possibly overlapping other lenses (right), can be highlighted.

5. Discussion

Our goal was to derive abstract representations of 3DCMs to reduce visual complexity for easier comprehension of urban structures in the context of interactive visualization. In addition, the derivation process must be automatic and work for data commonly used in the context of 3DCMs.

Our technique creates a number of discrete *LOA*s that hide a large number of unimportant single buildings while maintaining landmarks necessary for orientation and navigation. Clearly, our results present an abstraction of the input 3DCM. It works on standard input data, such as extruded buildings, CAD-based buildings and road network data, and runs automatically.

It can be questioned, whether 3D block cells would be necessary at all, as opposed to simple 2D built-up areas with 3D landmark buildings placed on top of them. We argue that flat 2D polygons would disrupt the impression of a 3D visual model. In addition, showing the average height of approximated buildings allows the user to distinguish suburban areas with rather low building heights from central areas of the city.

Our approach to use block cells as 3D built-up areas to replace individual buildings does not create convincing results in cases of sparsely built areas, such as the periphery of cities. There, another approach for aggregation is needed to account for empty spaces between buildings. Still, the technique works reasonably within central areas of cities, as building density is high enough here. Also, block cells are understandable units of cities, reflecting the higher-level mental image

people have. We successfully demonstrated further usability of the resulting geometry with different 3DCMs, in applications such as focus+context visualization and as a testbed for cognitive studies.

With multi-scale rendering and dynamic landmark highlighting, we present two approaches that exploit interactivity in visualization of 3DCMs beyond interactive exploration. By dynamically changing the scene geometry, we adapt complexity or point the user's focus to important landmark elements. Our current visualization still has potential for improvements and many parameters are chosen intuitively instead of being backed up by user studies. Still, we believe the combination of cartographic principles with the capabilities of real-time 3D visualization is promising and suggest to do further research this direction.

6. Conclusion & Outlook

As city models become ever more complete, it becomes necessary to apply principles of abstraction to their visualization in order to use them to communicate spatial information efficiently. We present here a method to create abstract representations for 3DCMs by deriving abstracted building blocks from the infrastructure network. By including non-building areas, such as wooded areas, lakes, and rivers, further important elements of 3DCMs are integrated. A number of applications show how to cope with dynamic changes of view in 3D visualization and how to communicate routes based on abstract representations, illustrating the feasibility of our approach.

Many directions open up for further research. One major area is the improvement of existing techniques to deal with current shortcomings, such as how to handle the representation of sparsely built residential areas, how to correct undesired displacement of landmark buildings, and how to mediate transparency issues when blending between representations.

In addition to incorporating existing measures for saliency, we would like to experiment with the detection of visual landmarks assisted by the GPU using image information metrics. Also, it should be possible to compute the "best" view for a single landmark, allowing to turn this side to the user dynamically when moving through the 3DCM. This visualization principle can be observed frequently on tourist maps, where it facilitates identification of landmarks.

Furthermore, abstract representations can be used as a more coarse reference frame for thematic data. As an example, data available on a per-house basis could easily be aggregated and visualized as block cells, using color or height. Coloring and styling elements of 3DCMs, including abstract building, vegetation, and water

features, could be optimized and formalized using styled layer descriptors (SLD) or symbology encoding standards (Neubauer and Zipf, 2007; OGC, 2006b).

A very important step for further research is to design and conduct user studies to evaluate the feasibility of our approach. Apart from the general acceptance of the presented *LOA*s, it will be interesting if and how users do understand scene elements presented simultaneously in different degrees of abstraction. Possibly, additional visual cues have to be integrated to further communicate this.

Acknowledgments

This work has been funded by the German Federal Ministry of Education and Research (BMBF) as part of the InnoProfile research group "3D Geoinformation" (www.3dgi.de). We are grateful for helpful comments from four anonymous reviewers as well as from our colleagues. We also thank Benjamin Richter for providing parts of the implementation.

References

Akenine-Möller, T., Haines, E., 2002. Real-Time Rendering, 2nd Edition. AK Peters, Ltd. Natick.

Anders, K.-H., 2005. Level of Detail Generation of 3D Building Groups by Aggregation and Typification. In: Proceedings of 22nd International Cartographic Conference.

Barrault, M., Regnauld, N., Duchêne, C., Haire, K., Baeijs, C., Demazeau, Y., Hardy, P., Mackaness, W., Ruas, A., Weibel, R., 2001. Integrating Multi-Agent, Object-Oriented and Algorithmic Techniques for Improved Automated Map Generalization. In: Proceedings of the 20th International Cartographic Conference. pp. 2110–2116.

Berg, M., Kreveld, M., Overmars, M., Schwarzkopf, O., 2000. Computational Geometry: Algorithms and Applications. Springer.

Bier, E. A., Stone, M. C., Pier, K., Fishkin, K., Baudel, T., Conway, M., Buxton, W., DeRose, T., 1994. Toolglass and Magic Lenses: The See-Through Interface. In: Proceedings of the Computer Human Interface (CHI) conference. ACM, pp. 445–446.

Cecconi, A., Galanda, M., 2002. Adaptive Zooming in Web Cartography. In: Computer Graphics Forum. Vol. 21. Blackwell Publishing, Inc, pp. 787–799.

Chang, R., Wessel, G., Kosara, R., Sauda, E., Ribarsky, W., 2007. Legible Cities: Focus-Dependent Multi-Resolution Visualization of Urban Relationships. IEEE Transactions on Visualization and Computer Graphics 13, 1169–1175.

Cohen, J., Manocha, D., 2005. Model Simplification. Visualization Handbook, Elsevier, 393–411.

Danciger, J., Devadoss, S., Mugno, J., Sheehy, D., Ward, R., 2009. Shape Deformation in Continuous Map Generalization. GeoInformatica 13, 203–221.

Döllner, J., 2007. Non-Photorealistic 3D Geovisualization. Springer, Ch. 16, pp. 229–240.

Döllner, J., Buchholz, H., Lorenz, H., November 2006. Ambient Occlusion - ein Schritt zur realistischen Beleuchtung von 3D-Stadtmodellen. GIS - Zeitschrift für Geoinformatik, 7–13.

Elias, B., Paelke, V., 2008. User-Centered Design of Landmark Visualizations. In: Meng, L., Zipf, A., Winter, S. (Eds.), Map-based Mobile Services. Springer, pp. 33–56.

Forberg, A., 2007. Generalization of 3D Building Data Based on a Scale-Space Approach. ISPRS Journal of Photogrammetry and Remote Sensing 62 (2), 104 – 111.

Glander, T., Döllner, J., 2007. Techniques for Generalizing Building Geometry of Complex Virtual 3D City Models. In: 2nd International Workshop on 3D Geo-Information: Requirements, Acquisition, Modelling, Analysis, Visualisation. Springer, pp. 381–400.

Glander, T., Peters, D., Trapp, M., Döllner, J., 2009. 3D Wayfinding Choremes: A Cognitively Motivated Representation of Route Junctions in Virtual Environments. In: Proceedings of the 12th AGILE International Conference on GI Science. Springer, pp. 407–427.

Grabler, F., Agrawala, M., Sumner, R. W., Pauly, M., 2008. Automatic generation of tourist maps. ACM Trans. Graph. 27 (3), 1–11.

Hoppe, H., 1998. Smooth View-Dependent Level-of-Detail Control and Its Application to Terrain Rendering. In: Proceedings of the conference on Visualization'98. IEEE, pp. 35–42.

Kada, M., 2007. Scale-Dependent Simplification of 3D Building Models Based on Cell Decomposition and Primitive Instancing. In: Spatial Information Theory. Springer, pp. 222–237.

Klippel, A., Richter, K., Hansen, S., 2006. Wayfinding Choreme Maps. In: Visual Information and Information Systems. Springer, pp. 94–108.

Klippel, A., Richter, K.-F., Barkowsky, T., Freksa, C., 2005. The Cognitive Reality of Schematic Maps. In: Map-based Mobile Services. Springer, pp. 55–71.

Lamy, S., Ruas, A., Demazeau, Y., Jackson, M., Mackaness, W., Weibel, R., 1999. The Application of Agents in Automated Map Generalisation. In: Proceedings of the 19th International Cartographic Conference. pp. 160–169.

Landis, H., 2002. Production-Ready Global Illumination. RenderMan in Production (SIGGRAPH 2002): Course 16.

Lonergan, M., Jones, C. B., December 2001. An Iterative Displacement Method for Conflict Resolution in Map Generalization. Algorithmica 30 (2), 287–301.

Lynch, K., 1960. The Image of the City. MIT Press.

Maass, S., Trapp, M., Kyprianidis, J. E., Döllner, J., Eichhorn, M., Pokorski, R., Bäuerlein, J., Hesberg, H. v., October 2008. Techniques for the Interactive Exploration of High-Detail 3D Building Reconstruction Using the Example of Roman Cologne. In: Ioannides, M., Addison, A., Georgopoulos, A., Kalisperis, L. (Eds.), Proceedings of 14th International Conference on Virtual Systems and Multimedia (VSMM 2008). Archaeolingua, pp. 223–229.

MacEachren, A., 1995. How maps work. Guilford Press New York.

McMaster, R., Shea, K., 1992. Generalization in Digital Cartography. Association of American Geographers Washington, DC.

Möller, T., Haines, E., 1999. Real-Time Rendering. A. K. Peters, Ltd., Natick, MA, USA.

Neubauer, S., Zipf, A., October 2007. Suggestions for Extending the OGC Styled Layer Descriptor (SLD) Specification Into the Third Dimension - An Analysis of Possible Visualization Rules for 3D City Models. In: Rumor, M., Coors, V., Fendel, E. M., Zlatanova, S. (Eds.), Urban and Regional Data Management. UDMS Annual 2007. Routledge, pp. 133–142.

Nöllenburg, M., Merrick, D., Wolff, A., Benkert, M., 2008. Morphing Polylines: A Step Towards Continuous Generalization. Computers, Environment and Urban Systems 32 (4), 248–260.

OGC, 2006a. Candidate OpenGIS CityGML Implementation Specification.

OGC, July 2006b. Symbology Encoding Implementation Specification, Version 1.1.0. Open Geospatial Consortium Inc.

Pajarola, R., Gobbetti, E., 2007. Survey of Semi-Regular Multiresolution Models for Interactive Terrain Rendering. The Visual Computer 23, 583–605.

Rau, J.-Y., Chen, L.-C., Tsai, F., Hsiao, K.-H., Hsu, W.-C., 2006. LOD Generation for 3D Polyhedral Building Model. In: Advances in Image and Video Technology. Springer, pp. 44–53.

Raubal, M., Winter, S., 2002. Enriching Wayfinding Instructions with Local Landmarks. In: GIScience '02: Proceedings of the Second International Conference on Geographic Information Science. Springer, London, UK, pp. 243–259.

Sester, M., 2000. Generalization Based on Least Squares Adjustment. International Archives of Photogrammetry and Remote Sensing 33, 931–938.

Sester, M., Brenner, C., 2004. Continuous Generalization for Fast and Smooth Visualization on Small Displays. International Archives of Photogrammetry 35, 1293–1298.

Sorrows, M. E., Hirtle, S. C., 1999. The Nature of Landmarks for Real and Electronic Spaces. In: COSIT '99: Proceedings of the International Conference on Spatial Information Theory: Cognitive and Computational Foundations of Geographic Information Science. Springer, London, UK, pp. 37–50.

Trapp, M., Döllner, J., April 2008. Real-Time Volumetric Tests Using Layered Depth Images. In: K. Mania, E. R. (Ed.), Proceedings of Eurographics 2008. The Eurographics Association, pp. 235–238.

Trapp, M., Glander, T., Buchholz, H., Döllner, J., July 2008. 3D Generalization Lenses for Interactive Focus + Context Visualization of Virtual City Models. In: Proceedings of the 12th International Conference on IEEE Information Visualization. IEEE, pp. 356–361.

van Kreveld, M., 2001. Smooth Generalization for Continuous Zooming. In: Proceedings of 20th International Cartographic Conference. pp. 2180–2185.

Vinson, N., 1999. Design Guidelines for Landmarks to Support Navigation in Virtual Environments. In: Proceedings of the Computer Human Interface (CHI) conference. ACM, pp. 278–285.

Winter, S., Tomko, M., Elias, B., Sester, M., 2008. Landmark Hierarchies in Context. Environment and Planning B: Planning and Design 35 (3), 381–398.

Wolff, M., Asche, H., 2008. Geospatial Modelling of Urban Security: A Novel Approach with Virtual 3D City Models. In: Gervasi, O., Murgante, B., Laganà, A., Taniar, D., Mun, Y., Gavrilova, M. (Eds.), Computational Science and Its Applications - ICCSA 2008. Vol. 5072 LNCS of Lecture Notes in Computer Science. Springer, pp. 42–51.